**Study Guide for Midterm Exam**

**Comp 4952 2015**

Lecture notes: Introduction to HCI Part 1 and 2; Modules 1 and 2.

Labs: Lab assignments and issues discussed in lab.

Quizzes.

Readings: as specified in assignments and power-point presentations.

**Learning Outcomes / Questions**

1. What are the characteristics of a well-designed interface?

A well designed interface is:

* Controllable
* Comprehensible
* Predictable

For the user they should feel

* Competent
* Responsible
* Satisfied

1. What is usability?

* **Usability** is the ease of use and learnability of a human-made object. The object of use can be a software application, website, book, tool, machine, process, or anything a human interacts with.

1. What methodology is employed with respect to usability?

* a task-centered design prototyping methodology
* a user-centered methodology for the interface design

This means we will make the user interface work with how the user will do the task in real life, while at the same time allowing the user do everything they need to do.

Done in three stages:

* + Low fidelity design
  + Medium fidelity design
  + High fidelity design

1. Describe the methods and outcomes of low-fidelity prototype.

* Needs assessment
  + Is a study to figure out what the needs to do with the application in comparison to what they are doing now
* User task analysis
  + Is a study to figure out what the user is doing with the application and how they do them
* Usability study
  + Is a study to figure out if your design works with the user

1. Name four rules of interface design and describe them (Shneiderman).

* Strive for consistency
* Enable frequent users to use shortcuts
* Offer informative feedback
* Design dialog to yield closure
* Offer simple error handling
* Permit easy reversal of actions
  + Support internal locus of control
    - Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions
    - Design the system to make users the initiators of actions rather than the responders
* Reduce short-term memory load

1. Describe four usability heuristics proposed by Jakob Nielsen and give a short description.
   * **Visibility of system status**

* The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
* **Match between system and the real world**
* The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.
* Follow real-world conventions, making information appear in a natural and logical order.
* **User control and freedom**
* Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
* **Consistency and standards**
* Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
* **Error prevention**
  + Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
  + Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
* **Recognition rather than recall**
  + Minimize the user's memory load by making objects, actions, and options visible.
  + The user should not have to remember information from one part of the dialogue to another.
  + Instructions for use of the system should be visible or easily retrievable whenever appropriate.
* **Flexibility and efficiency of use**
  + Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users.
  + Allow users to tailor frequent actions.
* **Aesthetic and minimalist design**
  + Dialogues should not contain information which is irrelevant or rarely needed.
  + Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
* **Help users recognize, diagnose, and recover from errors**
  + Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
* **Help and documentation**
  + Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation.
  + Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

1. Describe the usability heuristics proposed by Jakob Nielsen that you used in your low-fidelity prototype design and justify them.

**Consistency and standards**

* Our application has the same view throughout

**Recognition rather than recall**

Our application has the same icons in the same places so that the user knows where to look for information

**Flexibility and efficiency of use**

Stateless design allows for the user to do anytask in any order giving them the greatest flexibility.

1. Indicate practical strategies for usability and describe them.

Test early

Determine goals of the application and build tasks off of it

Determine the user profile

Run sessions of testing

1. Consider Fitts’s law: MT = a + b \* log2 (2A/W + c). Explain it.
   1. is used to model the act of pointing by physically touching an object, or virtually by pointing to an object on a computer monitor using a pointing device (i.e. mouse).

In other word, large things are easier to click, and things that are closer to the mouse are easier to click. In addition things that are in the borders of the screen are infinitely easier to click

The equation yields the ID (Index of difficulty)

The ID increases by one unit for each doubling of amplitude and halving of width.

1. Describe how Fitts’s law can be used for interface design.

From this we can design our interface to make things of importance bigger so that they are easier to click, as well move commonly used shortcuts into the corners of the screen. This allows us to layout our screens to make the mouse closes to the next point the user will click.

Add contextual menus on click.

1. What are consequences and heuristics of Fitts’s law?

From this we can design our interface to make things of importance bigger so that they are easier to click, as well move commonly used shortcuts into the corners of the screen. This allows us to layout our screens to make the mouse closes to the next point the user will click.

Add contextual menus on click.

1. How did you apply Fitts’s law in your design?
2. Explain usability and aesthetics.

Usability is influenced by aesthetics

Give example of fancy ATM machine

1. How did you perform the user task analysis in your low-fidelity prototype design?
2. What is user modeling?

Using a decision model that takes into consideration users’ beliefs and intentions and goals

1. Consider the Bayesian theorem: P (H|X) = (P (X|H)\* P (H)) / P(X). Explain it.

An equation that predicts a future outcome based on a previously predicted outcome.

Conditional probabilities

* + H is the hypothesis
  + X is the data
  + P (H|X) is the posterior probability of H conditioned on X
  + P(H) is the prior probability of H
  + P(X|H) is the posterior probability of X conditioned on H
  + P(X) is the prior probability of X (or normalized constant)

1. Consider the Bayesian theorem: P (H|X) = (P (X|H)\* P (H)) / P(X). Solve a problem with a Bayesian theorem and draw the tree graph.
2. What is the difference between a stochastic and a deterministic system?

-Stochastic is random and pertaining to chance. (The outcome cannot be predicted

-A system is deterministic if the output is the same given the same input.

1. To what extent statistical models can be used for user modeling?

Creating adaptive and adaptable interface design

* Example Youtube highlights.

1. Give four differences between ASP and ASP.NET

ASP is a procedural langague

ASP.net is object oriented

ASP.net decouples the code from the interface (HTML, and CS are not inline)

ASP.net can be compiled, wheras ASP is always interpreted

ASP is difficult to reuse as it is into intermingled with the front code.

1. What were the challenges of ASP that were solved with ASP.NET?

ASP.net is now object oriented which helps with reusabilitiy

Decoupling of server side code from the front end enhances readability

Solved limited session state by using the view state capability

1. What are Web Forms and Web Services?

Webforms are pretty much the equivalent of windows forms. The difference is that each form is now a web page. It also handles server side controls. All of the controls of a web form are similar. Web services do not have a front end, and are used to expose functionality to an application via HTTP. The client who knows the protocol can call the service.

1. Explain Postbacks.

* A postback is doing an HTTP post to the same page. This is the technique to handling form data in web forms. This is a page level property.

1. What is happening to server-side control objects during postbacks?

Server side control objects are automatically populated during a postback. This state is saved and repopulated when the page is reloaded.

1. How do technologies ASP and ASP.NET maintain the state of all server-side controls during postbacks?

This is done via the view state. The view state is a hidden input field inside of your webpage automatically created by ASP.net. This input field is encoded NOT ENCRYPTED in base 64 as well as compressed. This viewstate allows the controls to automatcailly populate itself. This view state can also allow the developer to add extra information.

1. In what event Page.IsPostBack is used? Why?

Ispostback allows the developer to check if this is the first time the user has visited the page. An example of this usage would be to set default values in controls along with display a tutorial on how to use the web form.

1. Are refresh and postback the same thing? Why or why not?

Refresh and post back are not the same thing. Refresh resends whatever information was requested to get to the current page. Where postback will “POST” new information back to the current page.

1. Describe server-side controls.

Server-side controls are controls that are rendered into HTML form elements. These server-side controls also allow for properties, and methods that can be used in the C# code behind that gives the developer more flexibility. An example of a control is the text input control.

1. Explain the page life cycle.

After receiving a http request, the server will load the page. During the page load there will be life cycle events that the developer can inject their own code such as populating form data, or calculating results. These life cycle events include

* Page\_Init
* Page\_load
* LoadViewState
* SaveViewState

1. Explain the Page\_Init() event.

The page\_init event is when the page is first started, the controls are first initialized. If this is a postback, this event is not fired again.

1. Explain LoadViewState() event.

This event only happens on post back. The view state data that has been received will be used to populate the controls in the page.

1. Explain Page\_Load () event.

The page load happens every single time the page is requested. This is a good place to initialize/reset variables.

1. How are variable stored in the ViewState?
2. What are the advantages and disadvantages of storing variables in ViewState?

View state is not secure, and takes up bandwidth as it is sent every single request. It is beneficial to keep track of the state of your application.

1. What security issues are related to the ViewState?

The view state is stored in a input field in a base 64 encoded text.

1. In what format is the ViewState stored? Is the ViewState encrypted by default?

The view state is stored in a input field in a base 64 encoded.

1. If no variables are stored in the ViewState is the ViewState empty? Explain.

The view state will not be empty if no variables are stored in the view state because control state information is stored in there as well.

1. What is stored in the ViewState?

State information is stored inside of the view state along with any other information you wish to store.

1. What is the ControlState?

The control state is similar to the view state except that this information cannot be disabled.

1. If you are a developer, is it possible to disable the ViewState? What about the ControlState? Why?

Yes it is possible to disable the view state; however it is not possible to disable the control state. This is because the control requires the control state to function, if you disable it the thing won’t work.

1. If you are a user, is it possible to disable the ViewState? What about the ControlState? Why?

Although you are not supposed to disable the view state, and control state it is possible to disable the view state. The process involves using the web browser’s debugger to remove the hidden input field. This results in the http request not sending the hidden information effectively disabling the view state.

1. What is happening when a user requests a page from a remote server?

A HTTP get request is sent to the remote server, where the web server (IIS) receives the request. At this point IIS will spawn a new thread and begin execution of the requested page. After execution of the page, the rendered html result is sent back to the user through IIS.

1. Is it any client-side code executing in your assignment application? If so, how is the client-side code executed in ASP.NET?

Yes the validation is done client sided. The validation control injects some extra Javascript code to help with the validation so that the user does not need to do multiple http requests to the server.

1. What is the role of the line:

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

Page language sets the language to C#

Auto event wire up will automatically override any lifecycle events to a method.

Codefile the actual file that contains the server side logic.

Inherits is the class name.

1. What is the role of a default web form?

This webform will be launched if the user requests the root folder on IIS. This page effectively acts as the default web page.

1. What special folders are in ASP.NET and what is their role?

App\_Code stores classes and datasets

Bin folder stores the binary assembiles to the application.

Add\_Data stores the data for the application such as mdf, or mdb files.

App\_Theme folder stores the css files for the web site

App\_browser stores browser information files

App\_references stores references to outside webservices

Resource folders

1. What is the difference between web server controls and HTML controls?

Web server controls render HTML controls. Web server controls add extra logic and functionality that the code behind file can access and modify before rendering.

1. Explain the property IsPostBack. What is the role of property IsPostBack and how should be used?
2. Describe the validation controls. What control validators are available in ASP.NET and how should be used?
3. What is the role of the property Page. isValid and how should be used?
4. What is happening when the user hits the refresh button?
5. What is MVC and how is it implemented in ASP.NET MVC?
6. Explain in your own words and give an example of model, view and controller in ASP.NET MVC.
7. Explain the role of the Entity Framework. Draw a diagram to explain your answer.
8. What does EF allow developers to do?
9. What are the main (basic) functions implemented in EF?
10. Describe the three aspects of EF: conceptual model, storage model and mapping.
11. What is the cost of EF? (for more details: read <https://msdn.microsoft.com/en-us/library/cc853327(v=vs.110).aspx> )
12. What is the most expensive operation with the EF?
13. What EF design models are implemented in ASP.NET?
14. Discuss the Entity Lifecycle. Refer to the ObjectContext and EntityState.
15. What are DataAnnotations and what do they describe? Give an example.
16. What was the role of the Context class when you created a data model?
17. What was the role of the Initializer class? What did it inherited from?
18. What design model did you used in the ASP.NET MVC lab activity?
19. What are migrations?